

FIG. 1 (PRIOR ART)

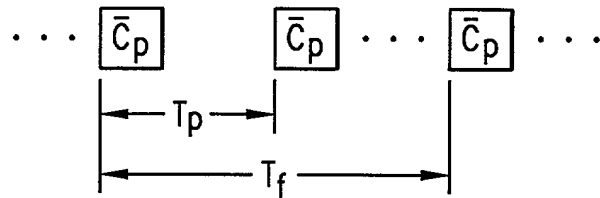


FIG. 2 (PRIOR ART)

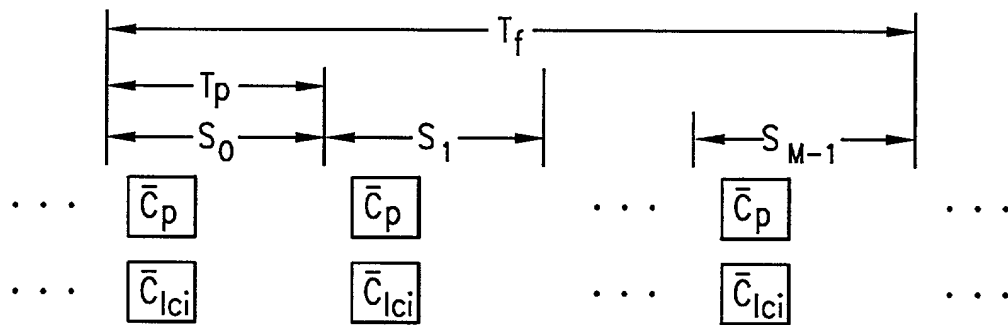


FIG. 3 (PRIOR ART)

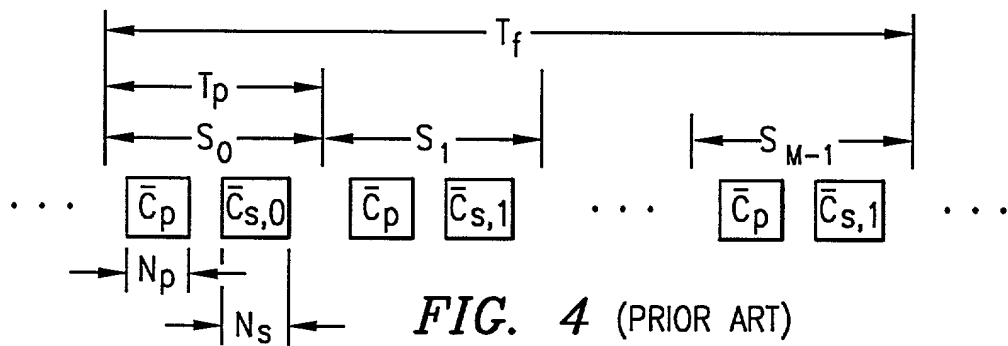


FIG. 4 (PRIOR ART)

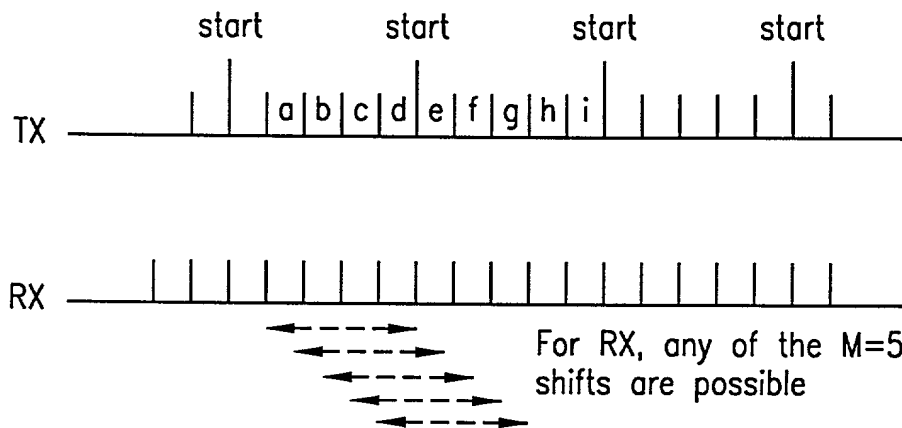


FIG. 5

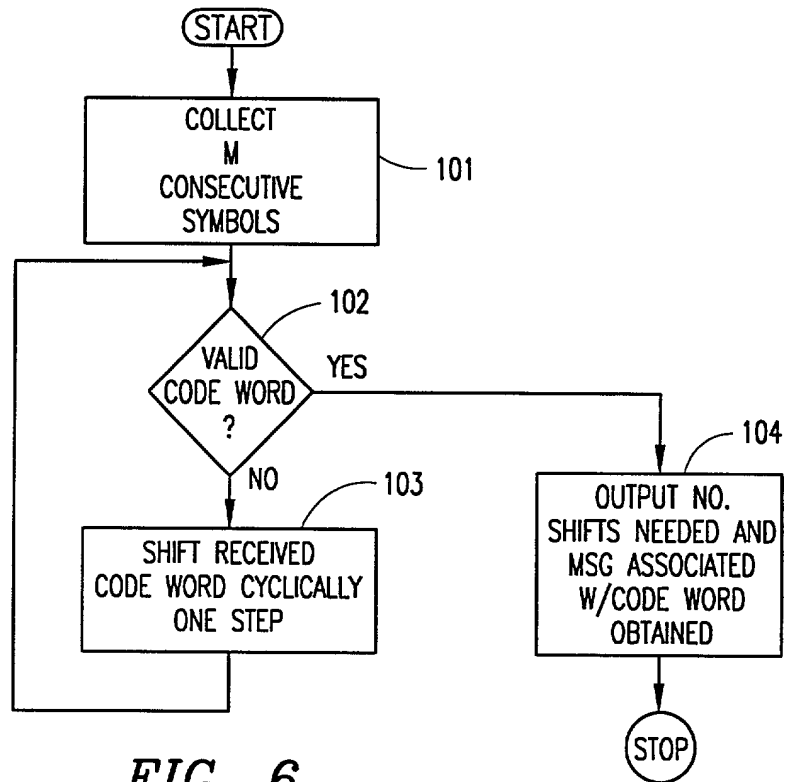


FIG. 6

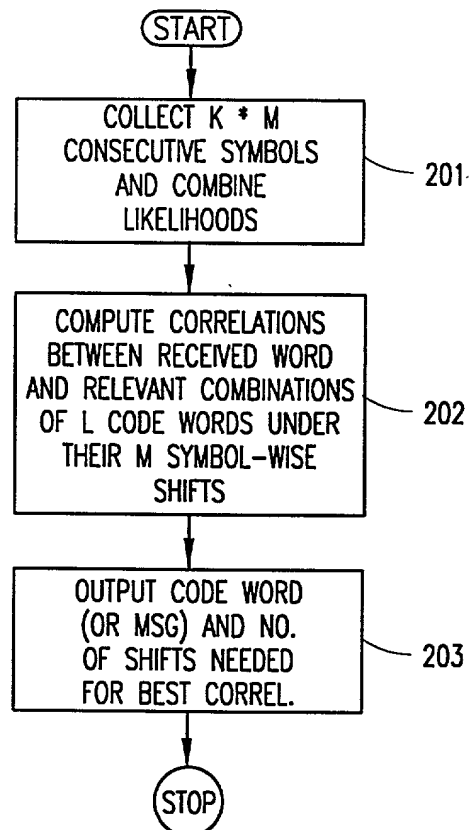


FIG. 7

3/9

FIG. 8

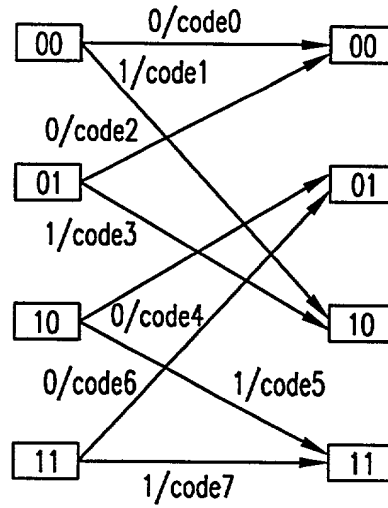


FIG. 9

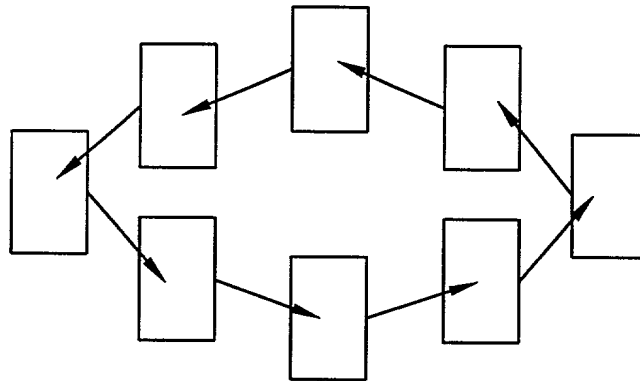
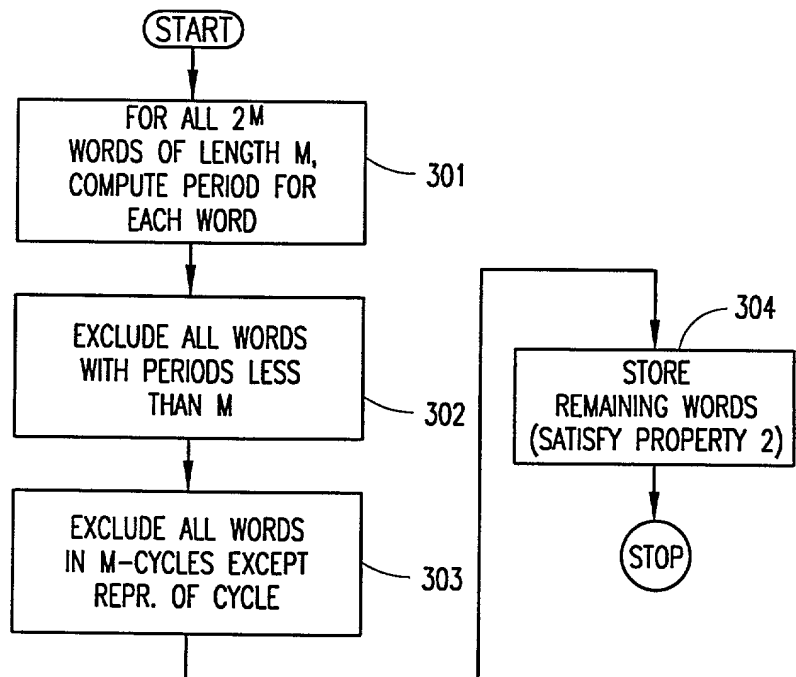
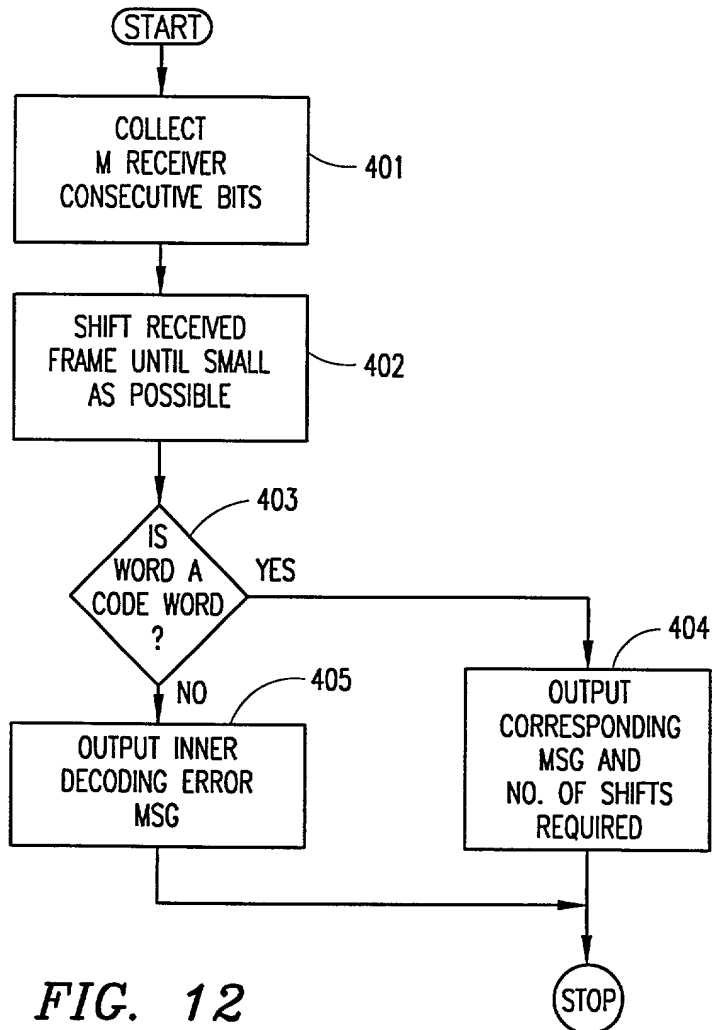


FIG. 10



00000->00000 , p=1, repr.: 00000
 00001->00010->00100->01000->10000->00001, p=5, repr.: 00001
 00011->00110->01100->11000->10001->00011, p=5, repr.: 00011
 00101->01010->10100->01001->10010->00101, p=5, repr.: 00101
 00111->01110->11100->11001->10011->00111, p=5, repr.: 00111
 01011->10110->01101->11010->10101->01011, p=5, repr.: 01011
 01111->11110->11101->11011->10111->01111, p=5, repr.: 01111
 11111->11111, p=1, repr.: 11111

FIG. 11**FIG. 12**

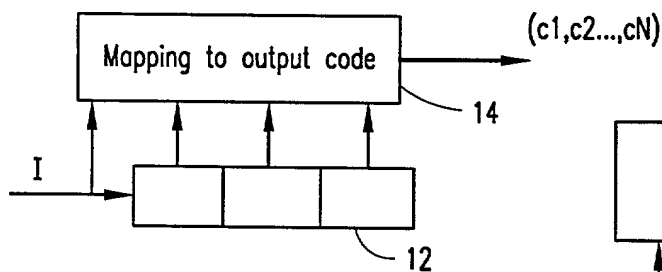
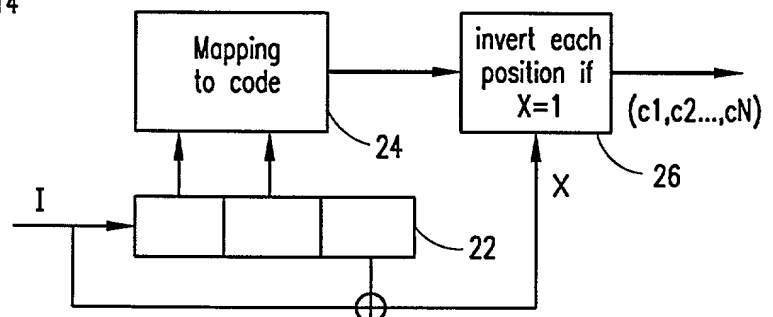
```

M=5;
code=[]; % will contain the set of code words
used=zeros(1,2^M-1); % keep track of words that are necessary to test
for i=1:2^M-1 % exclude all-zero sequence where p=1, always
if (used(i)==0) % if potential candidate, calculate period
x=i;
codeshifts=[i]; % first word in p-cycle
bad=0; % flag set if p<M
for j=1:M-1 % x==i after M shifts, of course
x=2*x; % shift
if (x>=2^M) x=x-2^M+1; end; % end around shift
if (x==i) bad=1; end; % is x==i after less than M shifts?
codeshifts=[codeshifts x]; % save smallest representative
used(codeshifts)=ones(size(codeshifts)); % and mark the shifts as used
end;

```

FIG. 13

M	Cardinality of code set
2	1
3	2
4	3
5	6
6	9
7	18
8	30
12	335
14	1161
16	4080

FIG. 14**FIG. 15A****FIG. 15B**

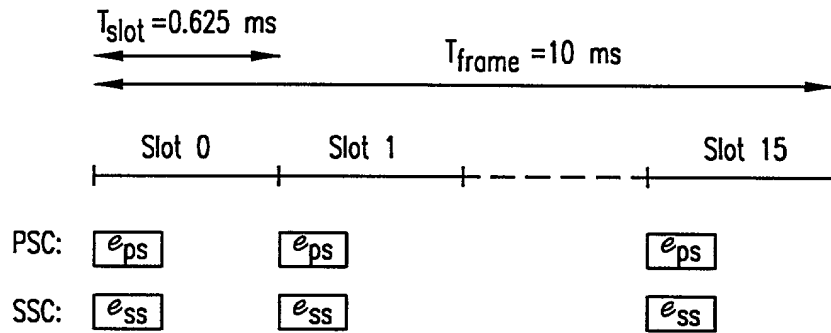


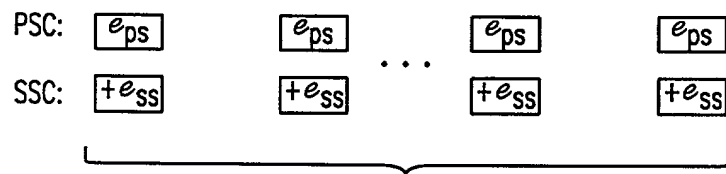
FIG. 16

Sync. Code	Code Type	Information		
		Slot Timing Indication (STI)	Frame Timing Indication (FTI)	Long Code Indication (LCI)
PSC	Orth. Gold	YES	NO	NO
SSC	Orth. Gold	-	NO	YES

FIG. 17

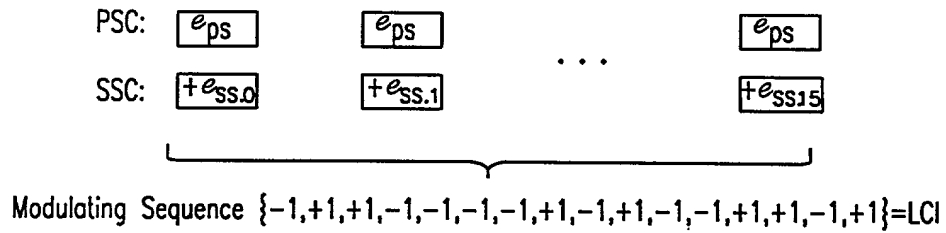
Sync. Code	Code Type	Information		
		Slot Timing Indication (STI)	Frame Timing Indication (FTI)	Long Code Indication (LCI)
PSC	Orth. Gold	YES	NO	NO
SSC	Orth. Gold	-	YES	YES

FIG. 18



Modulating Sequence $\{-1, +1, +1, -1, -1, -1, -1, +1, -1, +1, -1, -1, +1, +1, -1, +1\} = \text{FTI}$

FIG. 19

**FIG. 20**

STEP	METHOD		
	ARIB	PROPOSED 1	PROPOSED 2
1	MF→ST	MF→ST	MF→ST
2	CORR→LCI	CORR→LCI & FTI	CORR→FTI & LCI
3	CORR→LC & FT	(CORR→LC)	(CORR→LC)

FIG. 21

Parameter	Value
Chip Rate	4.096 Mc/s
Symbol Rate of the physical channel that carries BCCH	16 kSymbols/s
Frame Length	10 ms
Slots per Frame	16
Symbols per Slot	10
Chips per Symbol	256
Number of correlator units in MS	16
Number of coherently accumulated 256-chip correlations needed for sufficient noise/fading suppression	16
Number of non-coherently accumulated 256-chip correlations needed for sufficient noise/fading suppression	32
Number of long codes in the system	256
Long Code Grouping	1X256, 4x32 16x16, 32x4

FIG. 22

STEP	PROC	ARIB	METHOD 1	METHOD 2
2	CORR	Max 16x16x16= 4096, Avg 2048	Max 16x16x16= 4096, Avg 2048	16x16=256
	DELAY	Max 16 Frames, Avg 8 Frames	Max 16 Frames, Avg 8 Frames	1 Frame
3	CORR	16x16=256	No further correlations needed	No further correlations needed
	DELAY	16 Symbols=1.6 Slots	No further delay	No further delay
Total	CORR (avg) DELAY (avg)	2048+256=2304 8 Frames+1.6 Slots =8.1 Frames	2048 8 Frames	256+2048=256 1 Frame

FIG. 23A

STEP	PROC	ARIB	METHOD 1	METHOD 2
2	CORR	4x16=64	4x16=64	16x16=256
	DELAY	1 Frame	1 Frame	1 Frame
3	CORR	Max 16x32x32= 16384, Avg 8192	Max 32x32=1024, Avg 512	Max 32x32=1024, Avg 512
	DELAY	Max 1024 Symbols =102.4 Slots=6.4 Frames, Avg 3.2 Frames	Max 64 Symbols= 6.4 Slots=0.4 Frames, Avg 0.2 frames	Max 64 Symbols= 6.4 Slots=0.4 Frames, Avg 0.2 frames
Total	CORR (avg) DELAY (avg)	64+8192=8256 1 Frame+3.2 Frames=4.2 Frames	64+512=576 1 Frame+0.2 Frames=1.2 Frames	256+512=768 1 Frame+1.6 Slots =1.2 Frames

FIG. 23B

STEP	PROC	ARIB	METHOD 1	METHOD 2
2	CORR	16x16=256	16x16=256	16x16=256
	DELAY	1 Frame	1 Frame	1 Frame
Step 3	CORR	Max 16x16x32= 8192, Avg 4096	Max 16x32=512 Avg 256	Max 16x32=512 Avg 256
	DELAY	Max 512 Symbols= 51.2 Slots=3.2 Frames, Avg 1.6 Frames	Max 32 Symbols= 3.2 Slots, Avg 1.6 Slots=0.1 Frames	Max 32 Symbols= 3.2 Slots, Avg 1.6 Slots=0.1 Frames
Total	CORR (avg) DELAY (avg)	256+4096=4352 1 Frame+1.6 Frames=2.6 Frames	256+256=512 1 Frame+0.1 Frames=1.1 Frames	256+256=512 1 Frame+0.1 Frames=1.1 Frames

FIG. 23C

STEP	PROC	ARIB	METHOD 1	METHOD 2
2	CORR	Max 16x2x16=512, Avg 256		16x16=256
	DELAY	Max 2 Frames, Avg 1 Frame		1 Frame
3	CORR	Max 4x16x32=512, Avg 256	4x32=128	4x32=128
	DELAY	Max 128 Symbols= 12.8 Slots=0.8 Frames, Avg 0.6 Frames	32 Symbols=3.2 Slots=0.2 Frames	32 Symbols=3.2 Slots=0.2 Frames
Total	CORR (avg) DELAY (avg)	256+1024=1280 1 Frame+0.6 Frames=1.6 Frames	256+128=384 1 Frame+0.2 Frames=1.2 Frames	256+128=384 1 Frame+0.2 Frames=1.2 Frames

FIG. 23D